Math. Appl. **5** (2016), 123–145
DOI: 10.13164/ma.2016.09

ON FAREY TABLE AND ITS COMPRESSION FOR SPACE OPTIMIZATION WITH GUARANTEED ERROR BOUNDS

BISWAJIT PARIA, SANJOY PRATIHAR AND PARTHA BHOWMICK

Abstract. Farey sequences, introduced by such renowned mathematicians as John Farey, Charles Haros, and Augustin-L. Cauchy over 200 years ago, are quite well-known by today in *theory of fractions*, but its computational perspectives are possibly not yet explored up to its merit. In this paper, we present some novel theoretical results and efficient algorithms for representation of a Farey sequence through a *Farey table*. The *ranks* of the fractions in a Farey sequence are stored in the Farey table to provide an efficient solution to the *rank problem*, thereby aiding in and speeding up any application frequently requiring fraction ranks for computational speed-up. As the size of the Farey sequence grows quadratically with its order, the Farey table becomes inadvertently large, which calls for its (lossy) compression up to a *permissible error*. We have, therefore, proposed two compression schemes to obtain a *compressed Farey table* (CFT). The necessary analysis has been done in detail to derive the error bound in a CFT. As the final step towards space optimization, we have also shown how a CFT can be stored in a 1-dimensional array. Experimental results have been furnished to demonstrate the characteristics and efficiency of a Farey table and its compressed form.

1. Introduction

A *Farey sequence* F_n of *order* n is an (ordered) sequence of irreducible (simple) fractions starting from 0 and ending at 1 with denominators less than or equal to n [5, 7, 9]. Going by this definition, we have $F_1 = \left\langle \frac{0}{1}, \frac{1}{1} \right\rangle$, $F_2 = \left\langle \frac{0}{1}, \frac{1}{2}, \frac{1}{1} \right\rangle$, $F_3 = \left\langle \frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1} \right\rangle$, $F_4 = \left\langle \frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1} \right\rangle$, $F_5 = \left\langle \frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1} \right\rangle$, and so on. Such sequences are named after John Farey, who in 1816 conjectured that we can obtain F_n easily from F_{n-1} (Section 2.1), which was later proved by Cauchy. Interestingly, Charles Haros had published similar results in 1802, which were not known to Farey or Cauchy [9].

Although many interesting studies and research work related to Farey sequence are found in the *theory of fractions* [1,4,7,8,11,16–19,24], a limited work has been done so far from the algorithmic point of view. Two major problems concerned with the Farey sequence, which have been addressed in recent time, are the *rank problem* and the *order statistics problem*. The *rank* of a fraction x in a Farey sequence F_n is the number of fractions less than or equal to x in F_n. Given the order n, the rank problem is to find the rank of a given fraction in F_n, whereas

MSC (2010): primary 11K36, 11Y16, 11Z05.
Keywords: Farey sequence, Farey table, fraction rank, fraction error, rank problem, digital geometry, digital image processing.

the order statistics problem deals with finding the fraction in F_n with some given rank. Efficient solutions of these two problems may be seen in [20, 21].

Farey sequences are gradually finding interesting and practical uses in diverse areas of application, such as digital geometry, image processing, computer vision, and network modeling. In [14], it has been shown that fractions in a Farey sequence are related to straight-line covering of integer points in a convex polygon. Use of rational numbers to represent slopes of line segments joining integer points are often required in discrete- or digital-geometric applications of image processing, since the member fractions of a Farey sequence correspond to the *projection angles* [9, 13, 25]. However, the distribution of discrete angles corresponding to these fractions is not uniform. Hence, there have been studies related to their uneven distributions [26]. In a recent work, a relation between the Euclidean test mask and the Farey sequence has been established to detect the *Euclidean medial axis* of any discrete shape [10]. Farey sequences can also be related to *Farey graphs*, which were introduced in 1970's [15]. Such graphs possess interesting topological properties and can form a *network model* associated with complex systems, as shown recently in [27]. Farey sequence has also been employed to establish strict bounds for the order of the set of *equivalent resistances* for a circuit made of equal-valued resistors [12].

In another interesting work of recent time on randomness in signed-digit representation, the authors have studied how the class of random reals induced by such representation is related to the class of ordinary random reals [2]. The semantic name space of the representation is a rooted-tree-like structure, which is equivalent to *Stern-Brocot tree* or *Farey tree*, as the weights of the nodes follow *Stern's diatomic sequence* and so can be assigned as fractions in a Farey sequence. Very recently, in [3], some new characterization of *Sturmian words* in terms of the lexicographic order behavior of its factors has been established, which relates *word theory* to the *theory of fractions*, and hence relates to various other types of characterizations of geometric and combinatorial nature [13].

1.1. Our contribution

In this paper, we introduce the concept of *Farey table* from which the rank of a fraction can be obtained in constant time, and to be precise, only by a single memory access. It is a natural solution to obtain the fraction ranks almost instantaneously when they are queried very frequently in different procedural steps, such as comparing the slopes of two line segments with integer endpoints, checking the collinearity of three or more integer points, determining the type of turn (left or right) for three non-collinear integer points, etc. [22, 23]. It is worth mentioning here that runtime computation of the fraction ranks in a real-time application cannot serve the desired purpose, since even the best-known algorithm can compute the rank of a fraction in a time longer than $O(n^{2/3} \log^{1/3} n)$ [21].

In a Farey table designed by us, the row indices correspond to the fraction numerators, and the column indices to their denominators. The table entries are the ranks of the corresponding fractions in a Farey sequence, and hence a single memory access is needed to obtain the rank of a fraction from this precomputed table, as per the need of an application. However, as the order of the Farey

sequence goes high, the size of the table becomes inadvertently large, wherefore we resort to its compression so that the maximum error is within a permissible limit. We have proposed two different techniques of Farey table compression, given a *permissible error* ϵ^* in terms of fraction value. Both these compression techniques ensure that two different fractions of the concerned Farey sequence occupy the same position in the compressed Farey table (CFT) only if their difference does not exceed ϵ^*. In other words, for the rank problem mentioned earlier, when a rank query with a fraction x is applied to the CFT, the approximate rank of x returned from the CFT may correspond to a different fraction y, but guaranteeing that $x \in [y - \epsilon^*, y + \epsilon^*]$.

The rest of the paper is organized as follows. In Section 2, we show how an (uncompressed) Farey table can be generated by an optimal-time algorithm. In Sections 3 and 4, we discuss two different techniques of Farey table compression, given a value of the permissible error ϵ^*. For error analysis, we have derived an explicit formula for the maximum error in the rank of a fraction after table compression. We also show how the number of columns varies with the permissible error and the order of the Farey table. In Section 5, we explain how the compressed table can be stored in a 1-dimensional array, in order to optimize the storage space. With some of our test results presented in Section 6, we draw our concluding notes in Section 7.

2. Farey sequence and fraction rank

The Farey sequence of order n is given by $F_n = \left\langle \frac{i}{j} : \gcd(i,j) = 1 \wedge 0 \leqslant i \leqslant j \leqslant n \wedge j \neq 0 \right\rangle$. It has $\frac{3}{\pi^2}n^2 + O(n \log n) = \Theta(n^2)$ elements. A proof of this based on *Euler products* and *Riemann zeta function* is given in [9].

We define the *rank* of a fraction x w.r.t. F_n as $r_n(x) = |\{y : y \in F_n \wedge y \leqslant x\}|$. Note that rank of x is usually defined when $x \in F_n$ [7]. However, in this paper, we have defined, following [21], the rank of x regardless of its belongingness in F_n. Following the above definition, $r_n(x)$ can be written as

$$\begin{aligned}
r_n(x) &= \left| \left\{ \frac{i}{j} : \frac{i}{j} \leqslant x \wedge \gcd(i,j) = 1 \wedge 0 \leqslant i \leqslant j \leqslant n \wedge j \neq 0 \right\} \right| \\
\text{or, } r_n(x) - 1 &= \left| \left\{ \frac{i}{j} : \frac{i}{j} \leqslant x \wedge \gcd(i,j) = 1 \wedge 0 < i \leqslant j \leqslant n \right\} \right| \\
&= \left| \left\{ \frac{i}{j} : \frac{i}{j} \leqslant x \wedge 0 < i \leqslant j \leqslant n \right\} \right| - \\
&\quad \left| \left\{ \frac{i}{j} : \frac{i}{j} \leqslant x \wedge j \leqslant \left\lfloor \frac{n}{d} \right\rfloor \wedge \gcd(i,j) = d \geqslant 2 \right\} \right| \\
&= \sum_{i=1}^{n} \lfloor ix \rfloor - \sum_{d \geqslant 2} (r_{\lfloor \frac{n}{d} \rfloor}(x) - 1). \qquad (2.1)
\end{aligned}$$

Moving the second sum of (2.1) to its left-hand side, we get

$$\sum_{d \geqslant 1} (r_{\lfloor \frac{n}{d} \rfloor}(x) - 1) = \sum_{i=1}^{n} \lfloor ix \rfloor . \qquad (2.2)$$

For $n = 0$, we define $r_0(x) = 1$ for all $x \geqslant 0$, and it fits well into (2.2). It has been used in [21] to find the rank of a fraction in sub-linear time. Note that, (2.2) holds for both $x \in F_n$ and $x \notin F_n$.

2.1. Generation of Farey sequence

The *mediant* of two fractions $\frac{a}{b}$ and $\frac{c}{d}$ is defined as the fraction $\frac{a+c}{b+d}$, which can be shown to lie always between $\frac{a}{b}$ and $\frac{c}{d}$ [7]. Given the Farey sequence F_{n-1}, the next sequence F_n can be computed by introducing the mediant of every two consecutive fractions between them unless the denominator of the mediant exceeds n. The Farey sequence of any order can thus be generated starting with the Farey sequence of order 1, i.e., $\left\langle \frac{0}{1}, \frac{1}{1} \right\rangle$. To recursively generate F_n from F_{n-1}, it is required to check all pairs of adjacent fractions in F_{n-1}. This requires $\Theta(n^2)$ steps. Hence, the time complexity to generate a Farey sequence of order n is given by $T(n) = T(n-1) + \Theta(n^2) = \Theta(n^3)$.

The above algorithm can be improved by avoiding the concept of mediant and by generating the fractions using their adjacency relation. In particular, we use the following theorem.

Theorem 1. ([7], Chapter 4) *If $\frac{a_1}{b_1}$ and $\frac{a_2}{b_2}$ are two adjacent fractions in F_n, then the fraction in F_n lying next to $\frac{a_2}{b_2}$ is*

$$\frac{a_3}{b_3} = \frac{\left\lfloor \frac{n+b_1}{b_2} \right\rfloor a_2 - a_1}{\left\lfloor \frac{n+b_1}{b_2} \right\rfloor b_2 - b_1}. \tag{2.3}$$

A useful outcome of Theorem 1 is that the Farey sequence F_n of order n can be generated, starting with $\langle \frac{0}{1}, \frac{1}{n} \rangle$ in $\Theta(n^2)$ time. This is computationally optimal, since F_n contains $\Theta(n^2)$ elements. The ranks of the fractions in F_n are stored in the Farey table $\mathbf{F}_n$ while they are being generated, as shown in Algorithm 1. An instance of the Farey table generated by Algorithm 1 for $n = 20$ is shown in Table 1. Note that in this table, we also store the rank of a reducible fraction (i.e., a compound fraction, having the gcd of its numerator and denominator greater than 1), provided its numerator or denominator does not exceed n.

2.2. Fraction rank

Here we derive an explicit formula for $r_n(x)$, which is used to show its linearity with x.

Theorem 2. (Rank) *The rank of a fraction $x \in F_n$ is given by*

$$r_n(x) = \sum_{i=1}^{n} \left(\mu(i) \sum_{j=1}^{\lfloor \frac{n}{i} \rfloor} \lfloor jx \rfloor \right) + 1, \tag{2.4}$$

where $\mu(i)$ is the Möbius function *defined as*

$$\mu(i) = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if for some prime } p,\ p^2 | i \\ (-1)^k & \text{where } k \text{ is the number of distinct primes dividing } i. \end{cases}$$

Algorithm 1: Generation of Farey table

Input: Order of Farey sequence: `int` n
Output: Farey table of order n: $\mathbf{F}_n$

```
1  int a ← 0, b ← 1, c ← 1, d ← n, e ← 0, f ← 1, k, g
2  int F_n[n][n], rank ← 2, i ← 1
3  while i ⩽ n do
4  |   F_n[0][i] ← 1
5  |   i ← i + 1
6  end
7  F_n[1][n] ← 2
8  while e ≠ f do
9  |   k ← ⌊(n + b)/d⌋
10 |   e ← k · c − a,  f ← k · d − b   ▷ (2.3)
11 |   g ← gcd(e, f)
12 |   e ← e/g, f ← f/g
13 |   rank ← rank + 1,  i ← 1
14 |   while i · f ⩽ n do
15 |   |   F_n[e · i][f · i] ← rank
16 |   |   i ← i + 1
17 |   end
18 |   a ← c, b ← d, c ← e, d ← f
19 end
20 return F_n
```

Proof. Consider (2.2) and count the number of occurrences of each $r_i(x) - 1$, where $i = \lfloor \frac{n}{d} \rfloor$, $d \in [1, n]$. Observe that the number of d such that $\lfloor \frac{n}{d} \rfloor = i$ is $\left(\lfloor \frac{n}{i} \rfloor - \lfloor \frac{n}{i+1} \rfloor \right)$. Hence, we can write (2.2) as

$$\sum_{i \geqslant 1} \left(\left\lfloor \frac{n}{i} \right\rfloor - \left\lfloor \frac{n}{i+1} \right\rfloor \right) (r_i(x) - 1) \quad = \quad \sum_{i=1}^{n} \lfloor ix \rfloor \,. \tag{2.5}$$

Replacing n by $n - 1$, we have

$$\sum_{i \geqslant 1} \left(\left\lfloor \frac{n-1}{i} \right\rfloor - \left\lfloor \frac{n-1}{i+1} \right\rfloor \right) (r_i(x) - 1) \quad = \quad \sum_{i=1}^{n-1} \lfloor ix \rfloor \,. \tag{2.6}$$

Subtracting (2.6) from (2.5), we get

$$\sum_{i \geqslant 1} \left(\left\lfloor \frac{n}{i} \right\rfloor - \left\lfloor \frac{n-1}{i} \right\rfloor \right) (r_i(x) - 1) - \sum_{i \geqslant 1} \left(\left\lfloor \frac{n}{i+1} \right\rfloor - \left\lfloor \frac{n-1}{i+1} \right\rfloor \right) (r_i(x) - 1) = \lfloor nx \rfloor \,. \tag{2.7}$$

Table 1. The Farey table $\mathbf{F}_n$ for $n = 20$.

	$j=1$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$i=0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	129	65	43	32	26	22	18	16	14	12	11	10	9	8	7	6	5	4	3	2
2		129	87	65	52	43	37	32	29	26	24	22	20	18	17	16	15	14	13	12
3			129	98	78	65	56	49	43	39	35	32	30	28	26	25	23	22	21	19
4				129	104	87	74	65	58	52	47	43	40	37	34	32	31	29	27	26
5					129	108	93	81	72	65	60	54	50	46	43	41	38	36	33	32
6						129	112	98	87	78	70	65	61	56	52	49	45	43	42	39
7							129	114	101	91	83	76	69	65	62	57	53	51	48	44
8								129	116	104	95	87	80	74	68	65	63	58	55	52
9									129	118	106	98	90	84	78	73	67	65	64	59
10										129	119	108	100	93	87	81	77	72	66	65
11											129	120	110	102	96	89	85	79	75	71
12												129	121	112	104	98	92	87	82	78
13													129	122	113	105	99	94	88	86
14														129	123	114	107	101	97	91
15															129	124	115	108	103	98
16																129	125	116	109	104
17																	129	126	117	111
18																		129	127	118
19																			129	128
20																				129

Now, observe that

$$\left\lfloor \frac{n}{i} \right\rfloor - \left\lfloor \frac{n-1}{i} \right\rfloor = \begin{cases} 1 & \text{if } i|n \\ 0 & \text{otherwise.} \end{cases}$$

Hence, we can write (2.7) as

$$\sum_{i|n} (r_i(x) - 1) - \sum_{(i+1)|n} (r_i(x) - 1) \quad = \quad \lfloor nx \rfloor$$

or

$$\sum_{i|n} (r_i(x) - r_{i-1}(x)) \quad = \quad \lfloor nx \rfloor .$$

By applying *Möbius inversion formula* [9], the above equation reduces to

$$r_n(x) - r_{n-1}(x) \quad = \quad \sum_{i|n} \mu\left(\frac{n}{i}\right) \lfloor ix \rfloor . \tag{2.8}$$

Replacing n by j in (2.8) and summing up for $j = 1, 2, \ldots, n$, we get

$$\begin{aligned} \sum_{j=1}^{n} (r_j(x) - r_{j-1}(x)) \quad &= \quad \sum_{j=1}^{n} \sum_{i|j} \mu\left(\frac{j}{i}\right) \lfloor ix \rfloor \\ \text{or,} \quad r_n(x) - r_0(x) \quad &= \quad \sum_{j=1}^{n} \sum_{k|j} \mu(k) \left\lfloor \frac{j}{k} x \right\rfloor \text{ where } j = ik \\ \text{or,} \quad r_n(x) - 1 \quad &= \quad \sum_{j=1}^{n} \sum_{i|j} \mu(i) \left\lfloor \frac{j}{i} x \right\rfloor \text{ on replacing } k \text{ by } i \\ \text{or,} \quad r_n(x) - 1 \quad &= \quad \sum_{i,j \in S} \mu(i) \left\lfloor \frac{j}{i} x \right\rfloor \end{aligned} \tag{2.9}$$

where, $S = \{(i,j) : 1 \leqslant i \leqslant j \leqslant n \wedge i|j\}$.
Observe that $S = \{(i,ik) : 1 \leqslant i \leqslant ik \leqslant n\} = \bigcup_{i=1}^{n} \left\{(i,ik) : 1 \leqslant k \leq \left\lfloor \frac{n}{i} \right\rfloor\right\}$. Hence, (2.9) can be written as

$$\begin{aligned} r_n(x) - 1 &= \sum_{i=1}^{n} \left(\sum_{k=1}^{\left\lfloor \frac{n}{i} \right\rfloor} \mu(i) \left\lfloor \frac{ik}{i} x \right\rfloor \right) \\ \text{or,} \quad r_n(x) &= \sum_{i=1}^{n} \left(\mu(i) \sum_{k=1}^{\left\lfloor \frac{n}{i} \right\rfloor} \lfloor kx \rfloor \right) + 1, \end{aligned}$$

whence we get (2.4), thus completing the proof. □

2.3. Linearity of rank

Here we show that $r_n(x)$ has an asymptotically linear relation with x. Let f_n denote the number of fractions in F_n; i.e., $f_n = \max\{r_n(x) : x \in F_n\} = r_n\left(\frac{1}{1}\right)$. As mentioned in Section 2, $f_n = \Theta(n^2)$. For our work, we introduce the following lemma for f_n.

Lemma 2.1. *The number of fractions in F_n is given by*

$$f_n = \sum_{i=1}^{n} \left(\mu(i) \sum_{j=1}^{\left\lfloor \frac{n}{i} \right\rfloor} j \right) + 1.$$

Proof. Substituting $x = 1$ in Theorem 2 and observing that $f_n = r_n(1)$, we get the result. □

The above lemma is used to obtain the nature of variation of $r_n(x)$ with n and x, as stated in the following theorem.

Theorem 3. (Rank function) *For a given order n, the rank of a fraction x has an asymptotically linear relation with x. For a given fraction x, its rank has an asymptotically quadratic relation with n.*

Proof. From Theorem 2 and Lemma 2.1, we have

$$\begin{aligned} |(r_n(x) - 1) - (f_n - 1)x| &\leqslant \left| \sum_{i=1}^{n} \left(\mu(i) \sum_{j=1}^{\left\lfloor \frac{n}{i} \right\rfloor} \lfloor jx \rfloor \right) - \sum_{i=1}^{n} \left(\mu(i) \sum_{j=1}^{\left\lfloor \frac{n}{i} \right\rfloor} jx \right) \right| \\ &= \sum_{i=1}^{n} \left(\mu(i) \sum_{j=1}^{\left\lfloor \frac{n}{i} \right\rfloor} (jx - \lfloor jx \rfloor) \right) \\ &< \sum_{i=1}^{n} |\mu(i)| \left\lfloor \frac{n}{i} \right\rfloor \\ &\leqslant n \sum_{i=1}^{n} \frac{|\mu(i)|}{i} \end{aligned}$$

$$\begin{aligned} &\leqslant n\sum_{i=1}^{n}\frac{1}{i} \\ &\leqslant n(1+\log n). \end{aligned} \tag{2.10}$$

Therefore, we have the bounds on $r_n(x)$ as

$$(f_n-1)x - n(1+\log n) + 1 \leqslant r_n(x) \leqslant (f_n-1)x + n(1+\log n) + 1,$$

or

$$r_n(x) = \Theta(n^2 x), \text{ since } f_n = \Theta(n^2),$$

whence the proof. □

The above result is depicted through a 3D plot in Figure 1.

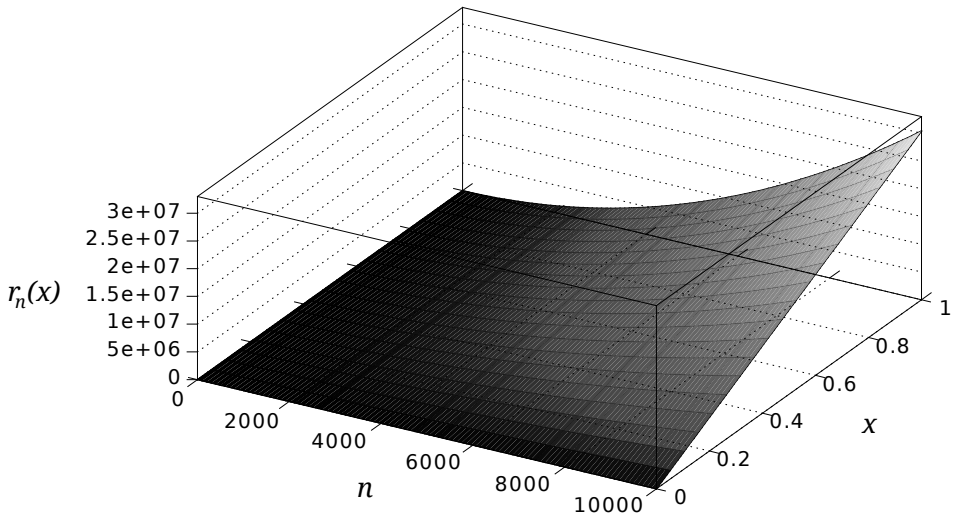

Figure 1. $r_n(x)$ versus n and x.

2.4. Maximum difference in ranks

Let $x \in F_n$ and $y \in F_n$, with $x > y$ and $\epsilon = x - y$. Then, using the lower and the upper bounds of $r_n(x)$ and $r_n(y)$ obtained from Theorem 3, it can be shown that

$$|(r_n(x) - r_n(y)) - (f_n - 1)(x - y)| \leqslant 2n(1 + \log n).$$

Hence, the difference in the ranks of x and y is given by

$$\delta \leqslant (f_n - 1)\epsilon + 2n(1 + \log n). \tag{2.11}$$

The above equation is used to estimate the maximum rank error for a specified fraction error during the Farey table compression, as explained in Section 3.

3. Farey table and a simple compression

We denote by $\mathbf{F}_n$ the Farey table corresponding to the Farey sequence, F_n. The table $\mathbf{F}_n$ stores the ranks of all fractions of F_n with the numerators indexed along the rows and the denominators indexed along the columns. Thus, the rank $r_n(\frac{i}{j})$ of a fraction $\frac{i}{j} \in F_n$ is stored in $\mathbf{F}_n[i][j]$. As an example, we have shown the Farey table for $n = 20$ in Table 1. Clearly, the rank of any fraction in F_n can be accessed from $\mathbf{F}_n$ in constant time.

3.1. Two-column compression

We can compress a Farey table $\mathbf{F}_n$ by merging two consecutive columns at a time. This is a straightforward technique, and an example of a compressed table for $n = 20$ is shown in Table 2. When two consecutive columns j and $j + 1$ are merged to a single column, the merged column contains the ranks of column $j+1$. Thus, after compression, $r_n(\frac{i}{j+1})$ provides an *approximate rank* for the fraction $\frac{i}{j}$, and its *exact rank* $r_n(\frac{i}{j})$ is no more available from the compressed table. The *average rank error* of merging of the elements of columns j and $j + 1$, denoted by $\delta^{(j)}_{\text{avg}}$, is defined as the average of differences between the approximate ranks and the exact ranks over all the elements in column j. That is,

$$\delta^{(j)}_{\text{avg}} = \frac{1}{j} \sum_{i=1}^{j} \left(r_n\left(\frac{i}{j}\right) - r_n\left(\frac{i}{j+1}\right)\right).$$

Similarly, the *average fraction error*, denoted by $\epsilon^{(j)}_{\text{avg}}$, in merging columns j and $j + 1$ is defined as the average of differences between the fractions corresponding to the approximate ranks and the fractions corresponding to the exact ranks for all elements in column j. That is,

$$\epsilon^{(j)}_{\text{avg}} = \frac{1}{j} \sum_{i=1}^{j} \left(\frac{i}{j} - \frac{i}{j+1}\right).$$

The *maximum rank error* in merging these two columns, denoted by $\delta^{(j)}_{\text{max}}$, is the maximum difference between the approximate and the exact ranks in column j. Similarly, the *maximum fraction error*, denoted by $\epsilon^{(j)}_{\text{max}}$, is defined as the maximum difference between the fraction corresponding to the exact rank and the fraction corresponding to the approximate rank over all elements in column j. That is,

$$\delta^{(j)}_{\text{max}} = \max_{1 \leqslant i \leqslant j} \left(r_n\left(\frac{i}{j}\right) - r_n\left(\frac{i}{j+1}\right)\right),$$

and

$$\epsilon^{(j)}_{\text{max}} = \max_{1 \leqslant i \leqslant j} \left(\frac{i}{j} - \frac{i}{j+1}\right).$$

We show here that the average rank error decreases as the index of the merged column increases. We require the following lemma (owing to the bijection $F_n \setminus (\frac{1}{2}, 1] \mapsto F_n \setminus [0, \frac{1}{2})$) to prove our result in Theorem 4.

Lemma 3.1. ([7]) *If two fractions x and y are equidistant from the middle element of F_n, i.e., $\frac{1}{2}$, then $x + y = 1$ and the sum of their ranks is $f_n + 1$.*

Theorem 4. (2-column rank error) *If the ranks in column j of $\mathbf{F}_n$ are replaced by the corresponding ranks in column $j+1$ of $\mathbf{F}_n$, then the average rank error is $\frac{f_n-1}{2j}$.*

Proof. Using Lemma 3.1, we have

$$\begin{aligned} r_n\left(\frac{i}{j}\right) + r_n\left(1-\frac{i}{j}\right) &= f_n+1 \\ \text{or,}\quad \mathbf{F}_n[i][j] + \mathbf{F}_n[j-i][j] &= f_n+1 \\ \text{or,}\quad \sum_{i=0}^{j}\left(\mathbf{F}_n[i][j] + \mathbf{F}_n[j-i][j]\right) &= (j+1)(f_n+1) \\ \text{or,}\quad \sum_{i=0}^{j}\mathbf{F}_n[i][j] + \sum_{i=0}^{j}\mathbf{F}_n[j-i][j] &= (j+1)(f_n+1) \\ \text{or,}\quad 2\sum_{i=0}^{j}\mathbf{F}_n[i][j] &= (j+1)(f_n+1) \\ \text{or,}\quad \sum_{i=0}^{j}\mathbf{F}_n[i][j] &= \frac{j+1}{2}(f_n+1). \end{aligned}$$

As $\mathbf{F}_n[i][j]$ assumes the value in $\mathbf{F}_n[i][j+1]$ after merging, the rank error of $\frac{i}{j}$ is $\mathbf{F}_n[i][j] - \mathbf{F}_n[i][j+1]$. Hence, the average rank error is given by

$$\begin{aligned} \delta_{\text{avg}}^{(j)} &= \tfrac{1}{j}\left(\textstyle\sum_{i=0}^{j}\mathbf{F}_n[i][j] - \sum_{i=0}^{j}\mathbf{F}_n[i][j+1]\right) \\ &= \tfrac{1}{j}\left(\textstyle\sum_{i=0}^{j}\mathbf{F}_n[i][j] - \sum_{i=0}^{j+1}\mathbf{F}_n[i][j+1] + \mathbf{F}_n[j+1][j+1]\right) \\ &= \tfrac{1}{j}\left(\tfrac{j+1}{2}(f_n+1) - \tfrac{j+2}{2}(f_n+1) + f_n\right) \\ &= \tfrac{f_n-1}{2j}. \end{aligned}$$

□

The above theorem signifies that the average rank error varies inversely with the index of the merged column. This gives an indication that more columns can be merged to a single column for higher column indices, which is used for generalized compression, explained in Section 4. To find the maximum fraction error in the two-column merging, observe that the error in fractions returning the same rank is $\frac{i}{j} - \frac{i}{j+1} = \frac{i}{j(j+1)}$. Hence, the error is maximum for $i = j$. Therefore, if ϵ^* denotes the *permissible error* in merging two columns, then

$$\frac{1}{j+1} \leqslant \epsilon^*, \text{ or, } j \geqslant \frac{1}{\epsilon^*} - 1. \tag{3.1}$$

The corresponding maximum error in their ranks can subsequently be obtained by (2.11).

For an example, see Table 2. As the pre-specified maximum permissible error is $\epsilon^* = 0.1$, we get $j \geqslant 1/0.1 - 1 = 9$ from (3.1), and hence, starting from column 9, every two columns have been merged at a time. Notice that, when two columns j and $j+1$ are merged, they are replaced by a single column comprising the elements of column $j+1$.

Table 2. Compressed Farey table after two-column compression for $n = 20$ and $\epsilon^* = 0.1$. See Table 1 for the Farey table before compression.

	$j=1$	2	3	4	5	6	7	8	9, 10	11, 12	13, 14	15, 16	17, 18	19, 20
$i=0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	129	65	43	32	26	22	18	16	12	10	8	6	4	2
2		129	87	65	52	43	37	32	26	22	18	16	14	12
3			129	98	78	65	56	49	39	32	28	25	22	19
4				129	104	87	74	65	52	43	37	32	29	26
5					129	108	93	81	65	54	46	41	36	32
6						129	112	98	78	65	56	49	43	39
7							129	114	91	76	65	57	51	44
8								129	104	87	74	65	58	52
9									118	98	84	73	65	59
10									129	108	93	81	72	65
11										120	102	89	79	71
12										129	112	98	87	78
13											122	105	94	86
14											129	114	101	91
15												124	108	98
16												129	116	104
17													126	111
18													129	118
19														128
20														129

4. Generalized compression

Compression of two columns at a time reduces the actual space requirement to half the original at the most, but cannot asymptotically reduce the space complexity of a Farey table. In this section, we show that compression of multiple columns at a time can asymptotically reduce the space complexity of the compressed Farey table (CFT) to a significant extent. The key observation leading to this compression scheme is that the maximum fraction error for compressing columns j and $j+1$ is the same as that for compressing columns $2j, 2j+1, 2j+2$, and also the same as that for compressing columns $4j, 4j+1, \ldots, 4j+4$, and so on. This, in fact, is in conformation with Theorem 4.

The procedure for merging multiple columns works in the following way. If the columns from j to j' are merged to a single column in a CFT, then the rth element of the resultant column is the median of the ranks of the rth elements of the columns j to j' that are merged. When the merging involves an even number of columns, there exist two median ranks, and the larger of these two is stored. This has been described in the procedure Merge of Algorithm 2. An illustration of the idea is shown in Fig. 2. Before explaining the generalized compression scheme, we introduce the following theorem to derive the maximum fraction error in this scheme.

Theorem 5. (Maximum fraction error) *Let* $S = \{2^k s, 2^k s+1, \ldots, 2^k s+2^{k+1}-1\}$, *where* $k \geqslant 0$, s *is even,* $2^k s \geqslant 2$, *and* $2^k s+2^{k+1}-1 \leqslant n$. *If* $\frac{i}{j}$ *is a fraction in* F_n *with* $j \in S$, *then the maximum fraction error, given by the difference between the fractions corresponding to its approximate rank and exact rank, is* $\epsilon_{\max} = \frac{1}{s+1} \leqslant \epsilon^*$.

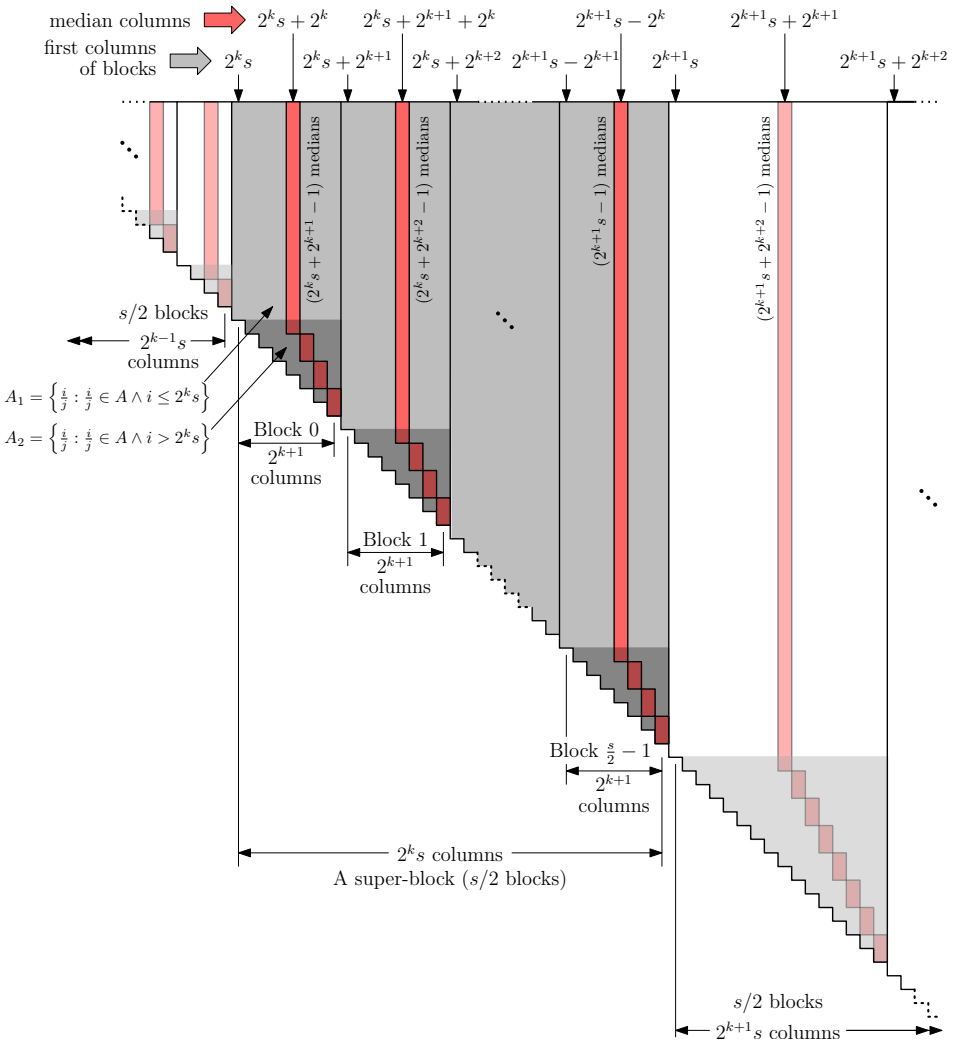

Figure 2. Illustration of generalized compression scheme.

Proof. We partition the set of fractions $A = \left\{\frac{i}{j} : j \in S, \frac{i}{j} \leqslant 1\right\}$ into two parts, $A_1 = \left\{\frac{i}{j} : \frac{i}{j} \in A \wedge i \leqslant 2^k s\right\}$ and $A_2 = \left\{\frac{i}{j} : \frac{i}{j} \in A \wedge i > 2^k s\right\}$, as shown in light gray and deep gray in Figure 2. If $\frac{i}{j}$ is a fraction in A_1, then the maximum fraction error associated with this fraction is given by

$$\epsilon_{\max(1)} = \max_{A_1}\left\{\left|\frac{i}{j} - \frac{i}{2^k s + 2^k}\right|\right\} \tag{4.1}$$

$$= \max_{A_1}\left\{i\left(\frac{1}{2^k s} - \frac{1}{2^k s + 2^k}\right), i\left(\frac{1}{2^k s + 2^k} - \frac{1}{2^k s + 2^{k+1} - 1}\right)\right\} \tag{4.2}$$

$$
\begin{aligned}
&= \max_{A_1}\left\{2^k s\left(\frac{1}{2^k s}-\frac{1}{2^k s+2^k}\right), 2^k s\left(\frac{1}{2^k s+2^k}-\frac{1}{2^k s+2^{k+1}-1}\right)\right\}\\
&= \max_{A_1}\left\{\frac{2^k}{2^k s+2^k}, \left(\frac{2^k-1}{2^k s+2^k}\right)\left(\frac{2^k s}{2^k s+2^{k+1}-1}\right)\right\}\\
&= \frac{2^k}{2^k s+2^k}, \text{ since } \frac{2^k s}{2^k s+2^{k+1}-1}<1 \text{ and } \frac{2^k-1}{2^k s+2^k}<\frac{2^k}{2^k s+2^k}\\
&= \frac{1}{s+1}. \qquad (4.3)
\end{aligned}
$$

Equations (4.1) and (4.2) indicate that the maximum fraction error for row i in any block (Figure 2) is simply the difference between the fraction of the median column and that of the first column lying in row i and in that block. And over all rows in the concerned block, we get the maximum error (see (4.3)) when the row is $i = 2^k s$. This fact is also used later to prove Theorem 6. Now, for each fraction $\frac{i}{j}$ in A_1, the maximum fraction error is given by

$$
\epsilon_{\max(2)} = \max_{A_2}\left\{\left|\frac{i}{j}-\frac{i}{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor}\right|\right\},
$$

since the median column in the ith row is $2^{k-1}s+2^k+\left\lfloor\frac{i}{2}\right\rfloor$

$$
= \max_{A_2}\left\{\left(\frac{i}{i}-\frac{i}{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor}\right), \left(\frac{i}{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor}-\frac{i}{2^k s+2^{k+1}-1}\right)\right\} \qquad (4.4)
$$

$$
= \max_{A_2}\left\{\left(\frac{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor-i}{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor}\right), \left(\frac{2^{k-1}s+2^k-1-\lfloor\frac{i}{2}\rfloor}{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor}\right)\left(\frac{i}{2^k s+2^{k+1}-1}\right)\right\}
$$

$$
= \max_{A_2}\left\{\left(\frac{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor-i}{2^{k-1}s+2^k+\lfloor\frac{i}{2}\rfloor}\right)\right\}
$$

$$
= \frac{2^{k-1}s+2^k+\lfloor\frac{2^k s+1}{2}\rfloor-2^k s-1}{2^{k-1}s+2^k+\lfloor\frac{2^k s+1}{2}\rfloor}
$$

$$
= \frac{2^{k-1}s+2^k+2^{k-1}s-2^k s-1}{2^{k-1}s+2^k+2^{k-1}s}
$$

$$
= \frac{2^k-1}{2^k s+2^k} < \frac{1}{s+1}
$$

and this completes the proof. □

Hence, we formulate the generalized compression scheme as follows. We merge two (consecutive) columns at a time for the columns with indices in $[s, 2s)$. The value of s is chosen as the smallest even integer greater than or equal to $\left\lceil\frac{1}{\epsilon^*}-1\right\rceil$, such that $\epsilon_{\max}(=\frac{1}{s+1})$ does not exceed the *permissible error*, i.e., ϵ^*, as mentioned earlier in Section 3.1. We get $\frac{s}{2}$ columns after compression of s columns. Next, we merge four columns at a time for the columns with indices in $[2s, 4s)$, thus getting again $\frac{s}{2}$ columns after compression of $2s$ columns. We continue this process by

Algorithm 2: GFTC (Generalized Farey Table Compression)

Input: $\mathbf{F}_n[\,][\,], n, \epsilon^*$
Output: $\mathbb{F}_n[\,][\,], c$
1 `int` $i,\ j,\ s \leftarrow 2\left\lfloor \frac{1}{2}\left\lceil \frac{1}{\epsilon^*}\right\rceil\right\rfloor,\ g \leftarrow 2$
2 `int` $c \leftarrow \min\{s, n\}$
3 **for** $i \leftarrow 1$ **to** $s-1$ **do**
4 | **for** $j \leftarrow i$ **to** $s-1$ **do**
5 | | $\mathbb{F}_n[i][j] \leftarrow \mathbf{F}_n[i][j]$
6 | **end**
7 **end**
8 **while** $s \leqslant n$ **do**
9 | $i \leftarrow s$
10 | **while** $(i + g - 1 < 2s) \wedge (i \leqslant n)$ **do**
11 | | $j \leftarrow \min\{i + g - 1, n\}$
12 | | Merge$(\mathbf{F}_n, i, j, \mathbb{F}_n, c)$
13 | | $c \leftarrow c + 1$
14 | | $i \leftarrow i + g$
15 | **end**
16 | $s \leftarrow 2 \cdot s, g \leftarrow 2 \cdot g$
17 **end**
18 **return** $\mathbb{F}_n, c$

Procedure Merge(`int` $\mathbf{F}_n[\,][\,]$, `int` l, `int` h, `int` $\mathbb{F}_n[\,][\,]$, `int` c)

1 `int` $i,\ m \leftarrow (l + h + 1)/2$
2 **for** $i \leftarrow 0$ **to** l **do**
3 | $\mathbb{F}_n[i][c] \leftarrow \mathbf{F}_n[i][m]$
4 **end**
5 **for** $i \leftarrow l + 1$ **to** h **do**
6 | $m \leftarrow (i + h + 1)/2$
7 | $\mathbb{F}_n[i][c] \leftarrow \mathbf{F}_n[i][m]$
8 **end**

merging 2^{k+1} columns at a time for the columns with indices in $[2^k s, 2^{k+1}s)$, till we have the required number of columns available for merging.

Figure 2 illustrates the generalized compression scheme, and Algorithm 2 shows the basic steps. Using Theorem 5, it first computes the column index s from which the merging should start (Step 1). Notice that the computation of s always yields its value as the smallest even integer greater than or equal to $\left\lceil \frac{1}{\epsilon^*} - 1\right\rceil$, as explained earlier. The variable g stores the block size, i.e., the number of columns that are to be merged next. The compressed Farey table is stored in the 2D array, $\mathbb{F}_n$. The variable c stores the index of the merged column, which is going to appear in $\mathbb{F}_n$. Line 2 ensures that the size of $\mathbb{F}_n$ doesn't exceed n in case the given permissible error ϵ^* is too small. The **for** loop (Lines 3–7) copies first $s-1$ columns of $\mathbf{F}_n$ to

Table 3. Compressed Farey table $\mathbb{F}_{20}$ ($n' = 8$) produced by Algorithm 2 from $\mathbf{F}_{20}$ (Table 1) for $\epsilon^* = \frac{1}{5}$.

	Super-block →			0		1		2
	Block →			0	1	0	1	0
	$j = 1$	2	3	4-5	6-7	8-11	12-15	16-20
$i = 0$	1	1	1	1	1	1	1	1
1	129	65	43	26	18	12	8	4
2		129	87	52	37	26	18	14
3			129	78	56	39	28	22
4				104	74	52	37	29
5				129	93	65	46	36
6					112	78	56	43
7					129	114	65	51
8						104	74	58
9						118	84	65
10						119	93	72
11						129	102	79
12							112	87
13							122	94
14							123	101
15							129	108
16								116
17								117
18								127
19								128
20								129

$\mathbb{F}_n$, since these columns are not to be merged. The outer **while** loop (Lines 8–17) performs the actual merging. In this loop, s marks the index of the first column of the first block in a sequence of blocks of columns. Each of these blocks has the same number of columns, and the columns in each block are merged to a single column (shown in red in Figure 2). This is performed in the inner **while** loop (Lines 10–15). The value of s is doubled after merging the requisite number of blocks in the inner **while** loop, and subsequently it takes the values $2s, 2^2s, \ldots, 2^k s$ (Line 16). The next block size g is also doubled when s is doubled. Line 11 takes care of proper compression of the last block, which may have fewer columns than the current value of g, depending on the value of n.

Table 3 shows a small-yet-representative example of the compressed Farey table $\mathbb{F}_{20}$ produced by Algorithm GFTC from $\mathbf{F}_{20}$ with $\epsilon^* = 0.2$. The starting index s is first computed to be 4. Hence, the first three columns are simply copied to $\mathbb{F}_{20}$, as they cannot be merged. This is performed in the **for** loop (Lines 3–7). After this, the **while** loop is executed (Lines 8–17). In the first iteration of this loop, two blocks of $g = 2$ columns each, starting from $s = 4$, are merged. That is, Columns 4 and 5 of $\mathbf{F}_{20}$ are merged to Column 4 in $\mathbb{F}_{20}$, and Columns 6 and 7 to Column 5. Next, on getting doubled, the value of s becomes 8 and that of g becomes 4. So, in the 2nd iteration, two blocks of four columns each, starting from $s = 8$, are merged to Columns 6 and 7 of $\mathbb{F}_{20}$. After this, s and c again get doubled to assume the values 16 and 8 respectively. This is the last iteration where, in the inner **while** loop, j gets the value 20 in Line 11, wherefore, in the last block, Columns 16–20 of $\mathbf{F}_{20}$ are merged to Column 8 in $\mathbb{F}_{20}$. As a result, from twenty columns of $\mathbf{F}_{20}$, we get eight columns in $\mathbb{F}_{20}$.

We have now the following theorem on the maximum fraction error in a particular row, for a given permissible error, ϵ^*.

Theorem 6. (Row error) *In the merging of columns with indices from $2^k s$ to $2^k s + 2^{k+1} - 1$, the maximum fraction error in row i is at most $\frac{i\epsilon^*}{2^k s}$, where ϵ^* is the permissible error in the generalized compression.*

Proof. From the proof of Theorem 5 (Equations (4.1) and 4.2), it is evident that for each row $i \in \{1, 2, \ldots, 2^k s\}$, the maximum error in a particular block occurs between the first and the last fractions at that row in the concerned block. Hence, the maximum error at row i is given by

$$\epsilon^{(i)}_{\max(1)} = i\left(\frac{1}{2^k s} - \frac{1}{2^k s + 2^k}\right) = \frac{i}{2^k s(s+1)} = \frac{i\epsilon_{\max}}{2^k s} \leqslant \frac{i\epsilon^*}{2^k s}.$$

Now, by (4.4), the maximum error at row $i \in \{2^k s + a : a = 1, 2, \ldots, 2^{k+1} - 1\}$ is given by

$$\epsilon^{(i)}_{\max(2)} = \frac{i}{i} - \frac{i}{2^{k-1}s + 2^k + \lfloor \frac{i}{2} \rfloor} = 1 - \frac{2^k s + a}{2^{k-1}s + 2^k + \lfloor \frac{2^k s + a}{2} \rfloor} = 1 - \frac{2^k s + a}{2^k s + 2^k + \lfloor \frac{a}{2} \rfloor}. \tag{4.5}$$

As $k \geqslant 0$ and $s \geqslant 2$, we can show that

$$\frac{2^k s + a}{2^k s + 2^k + \lfloor \frac{a}{2} \rfloor} \quad > \quad \frac{2^k s}{2^k s + 2^k}.$$

Hence, from (4.5),

$$\epsilon^{(i)}_{\max(2)} < 1 - \frac{2^k s}{2^k s + 2^k} = \frac{1}{s+1} = \epsilon_{\max}.$$

So, for $2^k s < i < 2^k s + 2^{k+1}$, we get $\frac{i}{2^k s} > 1$, or, $\epsilon^{(i)}_{\max(2)} < \epsilon_{\max} < \frac{i\epsilon_{\max}}{2^k s} \leqslant \frac{i\epsilon^*}{2^k s}$, which completes the proof. □

Observe that the count of columns in the last block is a power of 2 if and only if the order n of $\mathbf{F}_n$ is of the form $2^t s - 1$, whence it admits a *full compression* (otherwise, it admits a *partial compression*). For example, for $n = 20$, with $\epsilon^* = \frac{1}{5}$, we have $s = \lceil \frac{1}{\epsilon^*} - 1 \rceil = 4$, and so the last block has five columns (Table 3). The full block size could be, in fact, $2^3 = 8$, and hence we could have compressed $2 \times 8 = 16$ columns in last $\frac{s}{2} = 2$ blocks, thus having a provision of compressing $16 - 5 = 11$ more columns without compromising with the maximum permissible fraction error, i.e., $\epsilon^* = \frac{1}{5}$. In particular, we have the following theorem on column count after running Algorithm GFTC.

Theorem 7. (CFT size) $\mathbb{F}_n$ *contains* $\Theta\left(\frac{\log n}{\epsilon^*}\right)$ *columns, and so its size is* $\Theta\left(\frac{n \log n}{\epsilon^*}\right)$.

Proof. Algorithm GFTC does not compress the columns indexed in $[1, s-1]$, but compresses the ones indexed in $[s, n]$, where $s = \lceil \frac{1}{\epsilon^*} \rceil - 1$. Let $k_{\max}$ be the largest integer such that $2^{k_{\max}+1} s - 1 \leq n$, i.e., $k_{\max} = \lfloor \log(\frac{n+1}{s}) \rfloor - 1$. Then, for $0 \leq k \leq k_{\max}$, the columns indexed in $[2^k s, 2^{k+1} s)$ are merged to just $\frac{s}{2}$ columns

(Figure 2). Hence, the columns indexed in $[s, 2^{k_{\max}+1}s)$ are merged to $\frac{s \cdot k_{\max}}{2}$ columns, and the remaining columns in $[2^{k_{\max}+1}s, n]$ to at most $\frac{s}{2}$. Thus, the total column count in $\mathbb{F}_n$ turns out to be at most

$$s - 1 + \frac{s}{2}\left(\left\lfloor \log\left(\frac{n+1}{s}\right)\right\rfloor - 1\right) + \frac{s}{2} = \Theta\left(\frac{1}{\epsilon^*} + \frac{\log(n\epsilon^*)}{\epsilon^*}\right) = \Theta\left(\frac{\log(n\epsilon^*)}{\epsilon^*}\right).$$

Since $\epsilon^* < 1$, we get $\log(n\epsilon^*) < \log n$, and so it simplifies to $\Theta\big(\frac{\log n}{\epsilon^*}\big)$. As a result, $\mathbb{F}_n$ has a size of $\Theta\big(\frac{n \log n}{\epsilon^*}\big)$. □

5. Storing the compressed farey table

The elements (fraction ranks) stored in a compressed Farey table occupy less than half the total space allocated in a 2D array. Further, with increasing value of ϵ^*, the value of s decreases and larger blocks of columns get merged, thereby increasing the unused space. Hence, to optimize the storage space, we store the table in a 1-dimensional array. The indexing from 2D to 1D works as follows.

We store the 2D array $\mathbb{F}_n$ to a 1D array, namely F_n, considering the elements of $\mathbb{F}_n$ in column-major order. We call the set of columns in $[2^k s, 2^{k+1}s)$, $k \geqslant 0$, of $\mathbf{F}_n$ as a *super-block*. Observe that the number of blocks in each super-block is $s/2$, and the number of columns in each block is 2^{k+1}. See Figure 2 and also Table 3. Two types of ranks are stored in the compressed table $\mathbb{F}_n$, and in F_n, thereof. The *first type* are for the fractions $\{\frac{i}{j} : 1 \leqslant j < s\}$, whose ranks are not compressed. The *second type* corresponds to the rest, i.e., $\{\frac{i}{j} : s \leqslant j \leqslant n\}$. We have the following theorem on the index of a fraction of either type in F_n.

Theorem 8. (1D index) *The approximate rank of* $\frac{i}{j}$ *is given by*

$$\tilde{r}_n(i/j) = \begin{cases} \mathrm{F}_n\left[\frac{j(j+1)}{2} + i - 1\right] & \text{if } j < s \\ \mathrm{F}_n\left[\frac{s^2+s-2}{2} + \frac{1}{4}s(3s+2)2^{k-1} + 2^k t(s+t+1) + i\right] & \text{otherwise} \end{cases}$$

where, $k = \lfloor \log(j/s) \rfloor$, $t = \left\lfloor \frac{j - 2^k s}{2^{k+1}} \right\rfloor$, *and with the consideration that the first element has index* 0 *in* F_n.

Proof. A fraction $\frac{i}{j}$ of first type has its exact rank stored in F_n, and so its index in F_n is $\frac{j(j+1)}{2} + i - 1$.

We now derive the index of a fraction $\frac{i}{j}$ of second type. For this, we first determine the super-block k, which j belongs to, and it is given by $2^k s \leq j < 2^{k+1}s$, or, $k = \lfloor \log(j/s) \rfloor$. Next, in this super-block, we determine the particular block $t\,(\geqslant 0)$, which j belongs to. It is given by $2^k s + t2^{k+1} \leq j < 2^k s + (t+1)2^{k+1}$, or, $t = \left\lfloor \frac{j - 2^k s}{2^{k+1}} \right\rfloor$.

Now, the ranks of the fractions of first type all occur in $\mathbb{F}_n$ till index $2+3+\ldots+s = \frac{s^2+s-2}{2}$. To count the (rank-containing) cells in all super-blocks preceding the kth super-block in $\mathbb{F}_n$, we first count the cells in its hth super-block. Observe the following:

i) Any super-block, excepting possibly the last, has $s/2$ blocks.

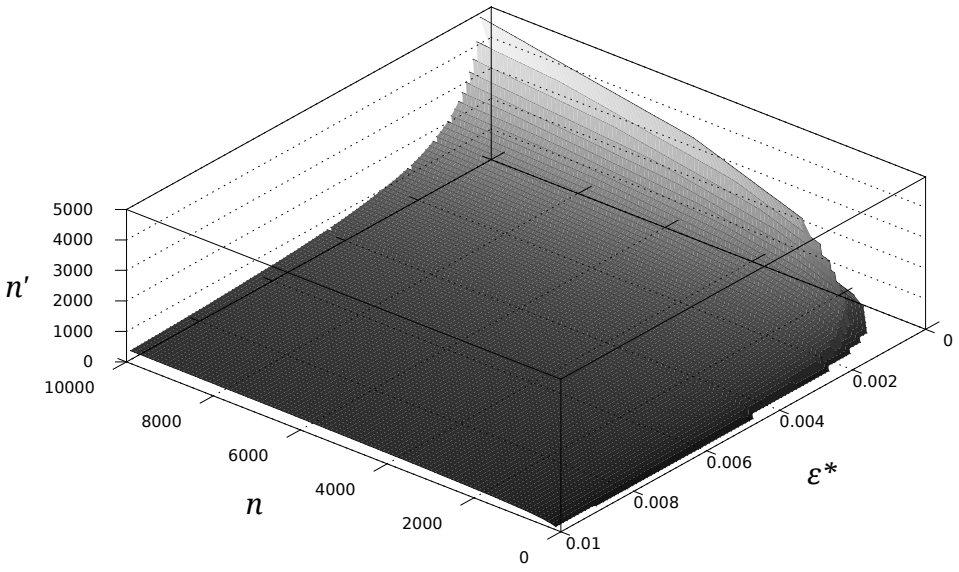

Figure 3. Number of compressed columns, n', versus maximum error ϵ^* and order n of Farey table.

ii) The count of cells in rth $(0 \leqslant r \leqslant s/2-1)$ block of hth super-block is $2^h s + (r+1)2^{h+1}$.

Hence, the count of cells in hth super-block is

$$c_h = \sum_{r=0}^{\frac{s}{2}-1} \left(2^h s + (r+1)2^{h+1}\right) = 2^{h-1}s^2 + \frac{1}{2}2^{h+1}\left(\frac{s}{2}+1\right)\frac{s}{2} = 2^{h-2}s(3s+2).$$

Thus, the count of cells up to $(k-1)$th super-block is $\sum_{h=0}^{k-1} c_h = \frac{1}{4}s(3s+2)2^{k-1}$. Reusing the two observations (i, ii) mentioned above, we also obtain the count of cells up to $(t-1)$th column of kth super-block as $\sum_{r=0}^{t-1}\left(2^k s + (r+1)2^{k+1}\right) = 2^k t(s+t+1)$. Since the number of cells preceding the cell of $r_n(\frac{i}{j})$ in tth block of kth super-block is i, we get the required result. □

6. Test results

We have implemented our algorithm and have studied its performance for different values of n and ϵ^*. Figure 3 shows a 3D plot of our experimental results. It can be inferred from this plot that the actual amount of compression obtained for our algorithm is in conformance with the theoretical result (Theorem 7). When ϵ^* is kept fixed, the number of compressed columns has a logarithmic dependence on n; and when n is fixed, the nature of dependency is hyperbolic.

It has been experimentally observed that for $n > 12$, the maximum value of $|r_n(x) - (f_n - 1)x|$ occurs at $x = \frac{1}{n},\ \frac{n-1}{n}$. Substituting x by $\frac{1}{n}$ or $\frac{n-1}{n}$, we get

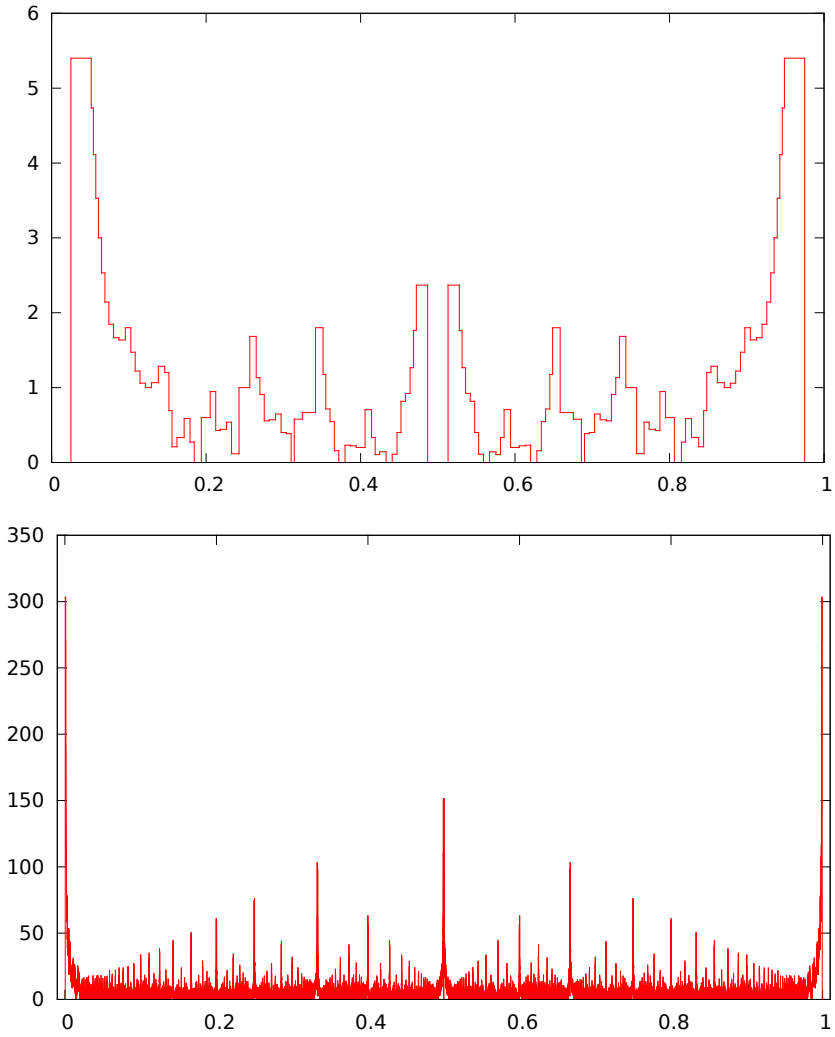

Figure 4. $|r_n(x) - (f_n - 1)x|$ versus x for $n = 20$ (top) and $n = 1000$ (bottom).

$|r_n(x) - (f_n - 1)x| = \frac{f_n - 1}{n} - 1 = \Theta(n)$, since $f(n) = \Theta(n^2)$ as mentioned in Section 2. This is better than the theoretical upper bound $n(1 + \log n)$ proved in Theorem 3 (see (2.10)), and this empirically obtained linear bound can be noticed in Figure 4.

From the plots shown in Figure 4, it may be noticed that the variation of $|r_n(x) - (f_n - 1)x|$ with x is not only linear, but it has a regular nesting pattern of increasing and decreasing nature. This pattern can be explained using a set of *Ford circles* [6], which also shows a similar nesting, as depicted in Figure 5. For each fraction $\frac{a}{b} \in F_n$, the corresponding Ford circle has radius $\frac{1}{2b^2}$ and center $(\frac{a}{b}, \frac{1}{2b^2})$. In Figure 5, the points of contact of the circles with the upper horizontal

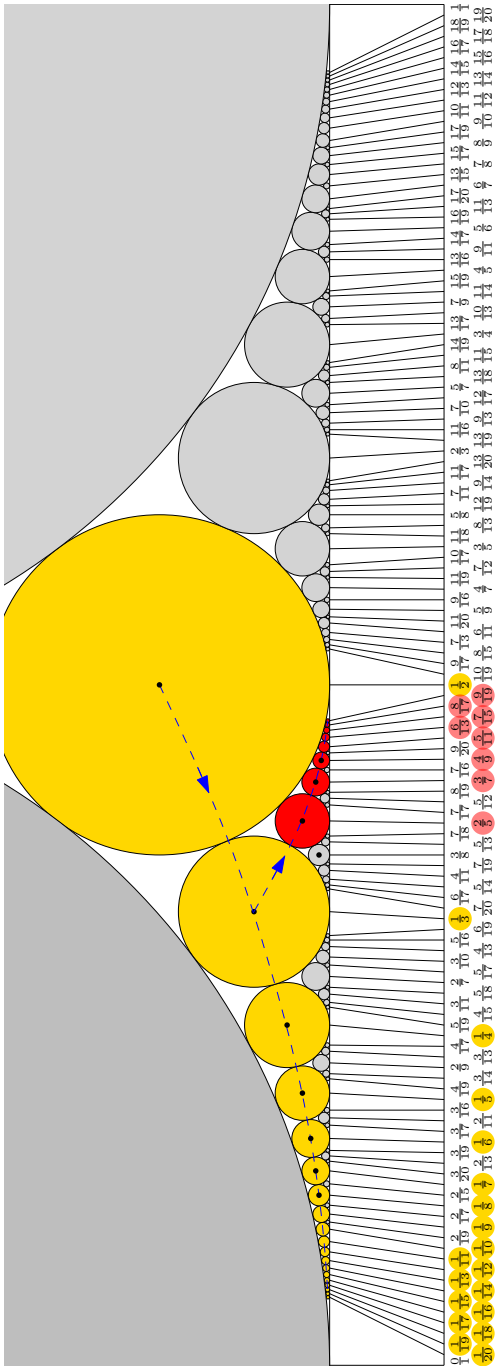

Figure 5. Ford circles for $n = 20$.

line are the fractions in F_n. On the lower horizontal line, their corresponding ranks are considered uniformly and in order. For clarity, the ranks are not shown, but the fractions are shown below the lower horizontal line. For correspondence, line segments are drawn joining the points of contact of the circles and their respective ranks. The slopes of these line segments correspond to $|r_n(x) - (f_n - 1)x|$. From these slopes, it can be noticed that in the initial part of the sequence, $|r_n(x) - (f_n - 1)x|$ is high, and its maximum value occurs at the beginning of the Farey sequence.

Also notice that there are repetitive patterns in the Ford circles formed due to the decreasing (or increasing) radius of the circles touching the same circle. For example, in Figure 5, the sequence of circles touching the largest circle centered at $(0, \frac{1}{2})$ corresponds to $\langle \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots, \frac{1}{20} \rangle$ in which the fractions are in decreasing order (highlighted in yellow). Similarly, the sequence of circles touching the 2nd largest circle from its left corresponds to $\langle \frac{1}{3}, \frac{2}{5}, \frac{3}{7}, \frac{4}{9}, \frac{5}{11}, \frac{6}{13}, \frac{7}{15}, \frac{8}{17}, \frac{9}{19} \rangle$ in which the fractions are in increasing order (highlighted in red). This phenomenon repeats for other circles too, thereby creating the repetitive patterns.

7. Conclusion

As mentioned in Section 1, a Farey table can be used to aid in and speed up various applications in which fraction ranks are frequently required for computational purpose. When the order n of the table is large, there arise space-related issues, which can be circumvented by a lossy compression of the table, especially for practical scenarios in which errors are pre-specified. Hence, after explaining the algorithm for Farey table generation in Section 2.1, we discussed the techniques for its compression in Sections 3 and 4. The compression techniques ensure a guaranteed error bound during compression. For a large order of the table, the compression algorithm proposed in Section 4 produces a table requiring $\Theta(n \log n)$ space instead of $\Theta(n^2)$, which can be stored in a 1D array, as shown in Section 5. As mentioned in Section 6, the bound of $|r_n(x) - (f_n - 1)x|$ is found to be $\Theta(n)$ from our test results, which indicates a possibility of improving our proven bound of $n(1 + \log n)$ in Theorem 3.

References

[1] M. Aigner, *Markov's Theorem and 100 Years of the Uniqueness Conjecture. A Mathematical Journey from Irrational Numbers to Perfect Matchings*, Springer, Switzerland, 2013.

[2] M. Archibald, V. Brattka, and C. Heuberger, *Randomness with respect to the signed-digit representation*, Fund. Inform. **83** (2008), 1–19.

[3] M. Bucci, A. D. Luca, and L. Q. Zamboni, *Some characterizations of sturmian words in terms of the lexicographic order*, Fund. Inform. **116** (2012), 25–33.

[4] C. Cobeli and A. Zaharescu, *The Haros–Farey sequence at two hundred years: A survey*, Acta Univ. Apulensis Math. Inform. **5** (2003), 1–38.

[5] J. Farey, *On a curious property of vulgar fractions*, Philos. Mag. **47** (1816), 385–386.

[6] L. R. Ford, *Fractions*, Am. Math. Monthly **45** (1938), 586–601.

[7] R. Graham, D. Knuth and O. Patashnik, *Concrete Mathematics*, Addison–Wesley, London, UK, 1994.

[8] S. B. Guthery, *A Motif of Mathematics: History and Application of the Mediant and the Farey Sequence*, Docent Press, Boston, 2010.

[9] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, Oxford University Press, New York, 1968.
[10] J. Hulin and E. Thiel, *Farey sequences and the planar Euclidean medial axis test mask*, in: P. Wiederhold and R. P. Barneva (eds.), Combinatorial Image Analysis, 13th international workshop, IWCIA 2009, Mexico, 2009, Lecture Notes in Computer Science **5852**, Springer, Berlin, 2009, 82–95.
[11] P. Kargaev and A. Zhigljavsky, *Asymptotic distribution of the distance function to the Farey points*, J. Number Theor. **65** (1997), 130–149.
[12] S. A. Khan, *Farey sequences and resistor networks*, Proc. Indian Acad. Sci., Math. Sci. **122** (2012), 153–162.
[13] R. Klette and A. Rosenfeld, *Digital Geometry: Geometric Methods for Digital Picture Analysis*, Morgan Kaufmann, San Francisco, 2004.
[14] H. Lee and R. Chang, *Covering grid points in a convex polygon with straight lines*, Int. J. Comput. Math. **42** (1992), 137–156.
[15] D. Matula and P. Kornerup, *A graph theoretic interpretation of fractions, continued fractions and the GCD algorithm*, in: R. C. Mullin and R. G. Stanton (eds.), Proc. 10th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica Publishing, 1979, p. 932.
[16] A. O. Matveev, *Relative blocking in posets*, J. Comb Optim. **13** (2007), 379–403, Corrigendum: `arXiv:math/0411026`.
[17] A. O. Matveev, *A note on boolean lattices and Farey sequences II*, Integers **8** (2008), Article A24, 8 pp.
[18] A. O. Matveev. *Neighboring fractions in Farey subsequences.* arXiv:0801.1981, 2010.
[19] E. H. Neville, *The Farey Series of Order 1025*, Cambridge University Press, 1950.
[20] C. E. Pătraşcu and M. Pătraşcu, *Computing order statistics in the Farey sequence*, D. Buell (ed.), Algorithmic Number Theory, 6th Tnternational Symposium, ANTS-VI, Burlington, VT, USA, 2004, Lecture Notes in Computer Science **3076**, Springer, 2004, 358–366. (2004).
[21] J. Pawlewicz and M. Pătraşcu, *Order statistics in the Farey sequences in sublinear time and counting primitive lattice points in polygons*, Algorithmica **55** (2009), 271–282.
[22] S. Pratihar and P. Bhowmick, *Vectorization of thick digital lines using Farey sequence and geometric refinement*, in: ICVGIP'10 Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing, 2010, 518–525.
[23] S. Pratihar and P. Bhowmick, *On applying the Farey sequence for shape representation in* $\mathbb{Z}^2$, Speech, Image and Language Processing for Human Computer Interaction: Multi-modal Advancements, Chapter 9, IGI Global, 2012, 172–190.
[24] M. Schroeder, *Fractions: Continued, Egyptian and Farey*, Number Theory in Science and Communication, Chapter 5, Springer Series in Information Sciences **7**, 2006.
[25] I. Svalbe and A. Kingston, *Farey sequences and discrete radon transform projection angles*, Electron. Notes Discrete Math. **12** (2003), 154–165.
[26] I. Svalbe and A. Kingston, *On correcting the unevenness of angle distributions arising from integer ratios lying in restricted portions of the Farey plane*, R. Klette and J. Zunic (eds.), Combinatorial Image Analysis, 10th International Workshop, IWCIA 2004, Auckland, New Zealand, 2004, Lecture Notes in Computer Science **3322**, Springer, 2004, 110–121.
[27] Z. Zhang and F. Comellas, *Farey graphs as models for complex networks*, Theor. Comput. Sci. **412**, 2011, 865–875.

Biswajit Paria, Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India
e-mail: biswajitsc@gmail.com

Sanjoy Pratihar, Department of Computer Science and Engineering, National Institute of Technology, Meghalaya, Shillong, India
e-mail: sanjoy.pratihar@gmail.com

Partha Bhowmick, Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India
e-mail: pb@cse.iitkgp.ernet.in, bhowmick@gmail.com